

## Задача А. Асхат и дерево

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

*Это интерактивная задача.*

Дано двоичное дерево поиска размера  $n$ , в каждой вершине есть значение от 1 до  $n$ . Петух по имени Асхат каждый день нумерует вершины числами от 1 до  $n$ , даёт вам номер корня  $r$  и просит вас найти номер вершины со значением  $x$ .

Вы можете совершать запросы — по номеру вершины узнать значение в ней и номера её детей. Пусть максимальное расстояние от корня до вершины в  $i$ -й день равно  $d$ . Тогда в  $i$ -й день вы можете совершить не более, чем  $d$  запросов. За все дни вы можете совершить не более, чем 70000 запросов.

Также вы можете менять детей у каждой вершины. Пусть в  $i$ -й день длина пути от корня до вершины с номером  $x$  равна  $s$ . Тогда вам в  $i$ -й день разрешено сделать не более, чем  $s$  запросов вида «установить у вершины новых детей». За все дни вы можете совершить не более, чем 70000 запросов этого типа.

### Протокол взаимодействия

В первой строке ввода даны числа  $n$  и  $q$  ( $1 \leq n \leq 2000$ ,  $1 \leq q \leq 2000$ ) — размер дерева и число запросов, соответственно.

Для каждого запроса даны числа  $r$  и  $x$  ( $1 \leq r, x \leq n$ ) — номер корня дерева и значение, вершину с которым требуется найти. Вы не сможете считать эти числа для следующего запроса, пока не дадите ответ на текущий.

Чтобы обратиться к вершине с номером  $i$  выведите «`val i`» в отдельной строке. В ответ даются три числа  $val$ ,  $L$  и  $R$  ( $1 \leq val \leq n$ ,  $0 \leq L, R \leq n$ ) — значение в этой вершине и номера левого и правого ребёнка, соответственно. В случае, если у вершины нет левого или правого ребёнка,  $L = 0$  или  $R = 0$ , соответственно.

Чтобы поменять детей вершины с номером  $i$  на вершины с номером  $L$  и  $R$  выведите «`change i L R`» в отдельной строке. Чтобы у вершины с номером  $i$  не было левого или правого ребёнка, выведите 0 вместо  $L$  или  $R$ , соответственно. После выполнения этого запроса граф может перестать быть двоичным деревом поиска.

Если искомое значение находится в вершине с номером  $i$  и вы совершили все нужные изменения, выведите «`confirm i`» в отдельной строке. После этого вершины перенумеруются, а на ввод будет дан новый запрос. Если на момент выполнения этого запроса граф не является двоичным деревом поиска, вы получите вердикт «Wrong answer».

## Задача В. Переворот

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан массив. Надо научиться обрабатывать два типа запросов.

- 1 L R - перевернуть отрезок [L, R]
- 2 L R - найти минимум на отрезке [L, R]

Напишите сплей-дерево.

### Формат входных данных

Первая строка файла содержит два числа  $n, m$ . ( $1 \leq n, m \leq 10^5$ ) Во второй строке находится  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ )- исходный массив. Остальные  $m$  строк содержат запросы, в формате описанном в условии. Для чисел L,R выполняется ограничение ( $1 \leq L \leq R \leq n$ ).

### Формат выходных данных

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

### Пример

стандартный ввод	стандартный вывод
10 7	3
5 3 2 3 12 6 7 5 10 12	2
2 4 9	2
1 4 6	2
2 1 8	
1 1 8	
1 8 9	
2 1 7	
2 3 6	

## Задача С. Динамический Лес

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф («link»).
2. Удалить ребро из графа («cut»).
3. По двум вершинам  $a$  и  $b$  вернуть длину пути между ними (или  $-1$ , если они лежат в разных компонентах связности) («get»).

Изначально граф пустой (содержит  $N$  вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

### Формат входных данных

Числа  $N$  и  $M$  ( $1 \leq N \leq 10^5 + 1$ ,  $1 \leq M \leq 10^5$ ) — количество вершин в дереве и, соответственно, запросов.

Далее  $M$  строк, в каждой строке команда («link» или «cut», или «get») и 2 числа от 1 до  $N$  — номера вершин в запросе.

### Формат выходных данных

В выходной файл для каждого запроса «get» выведите одно число — расстояние между вершинами, или  $-1$ , если они лежат в разных компонентах связности.

### Примеры

стандартный ввод	стандартный вывод
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	-1 1 -1 1
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	1 2 -1 1 -1 -1

## Задача D. Кратчайший путь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В некотором царстве, в некотором государстве есть несколько городов, соединённых двусторонними дорогами. При этом страна разбита на провинции, внутри каждой провинции можно добраться от любого города до любого другого, а между провинциями пути не существует. Внутри каждой провинции есть один выделенный город, который называется столицей этой провинции.

Компания «AloeVera» занимается пассажирскими перевозками в этом государстве. Компания перевозит людей от одного города до другого по кратчайшему пути. Однако, дорожная сеть постоянно развивается, поэтому периодически в государстве появляются новые дороги. Строительство новой дороги — очень важное событие, поэтому при появлении новой дороги все стремятся проехать именно по ней. Более точно, при добавлении новой дороги между городами  $u$  и  $v$  существует две возможности:

1. Города  $u$  и  $v$  находятся в разных провинциях. Тогда после добавления этой дороги эти две провинции объединяются, при этом столицей становится столица первой провинции (той, в которой до добавления находилась вершина  $u$ ).
2. Города  $u$  и  $v$  находятся в одной провинции. Тогда происходит перераспределение транспортных потоков. Услышав об открытии новой дороги, все продолжают добираться из столицы в другие города провинции по кратчайшему пути, но в случае, если таких путей существует несколько, то предпочтение отдается тому из них, на котором находится новая построенная дорога. После этого, если найдутся дороги, по которым никто ездить не будет, они навсегда удаляются из дорожной сети. Будьте внимательны: может так случиться, что по новой дороге никто ездить не будет, и тогда её не нужно добавлять вообще.

Иногда в государстве некоторые дороги приходят в негодность. При этом, если после удаления этой дороги некоторая часть провинции становится недостижимой из столицы, то она становится независимой провинцией, и её столицей назначается город, находящийся в этой провинции и являющийся концом удалённой дороги.

Основная же задача компании — сообщать людям о кратчайшем расстоянии между двумя какими-то городами. Ваша задача состоит в том, чтобы автоматизировать работу «AloeVera». При этом люди, сообщающие о разрушении дорог, иногда запаздывают или оперируют неверной информацией, и сообщают о разрушении дороги, которая уже была разрушена или вообще никогда не существовала.

Для решения поставленной задачи компания «AloeVera» использует следующий интерфейс:

- 1  $u$   $v$  — запрос об удалении дороги между городами  $u$  и  $v$ . Будьте внимательны, эта дорога могла быть разрушена ранее, или вообще никогда не существовать!
- 2  $u$   $v$   $w$  — запрос о добавлении новой дороги между городами  $u$  и  $v$ , время проезда по которой будет равно  $w$  ( $1 \leq w \leq 10^4$ ).
- 3  $u$   $v$  — запрос кратчайшего расстояния от города  $u$  до города  $v$ . В случае, если города находятся в разных провинциях, длина полагается равной  $-1$ .

Во всех запросах номера городов корректны, то есть  $1 \leq u, v \leq N$ , где  $N$  обозначает общее число городов в государстве.

### Формат входных данных

В первой строке входного файла содержится число  $N$  — количество городов в государстве. В следующей строке содержится число  $M$  — количество запросов. Далее в каждой строке содержится описание запроса. Описание соответствует условию.

$$1 \leq N \leq 50\,000, 1 \leq M \leq 100\,001.$$

## Формат выходных данных

Для каждого запроса типа 3 выведите единственное число: ответ на запрос.

## Примеры

стандартный ввод	стандартный вывод
5 8 2 1 3 10 2 1 2 3 1 1 2 3 1 3 2 1 3 9 2 2 3 1 3 1 5 3 1 3	10 -1 9
5 10 2 2 3 2 2 3 1 2 2 1 5 2 2 5 4 2 2 2 3 2 2 1 5 2 1 3 1 2 3 1 1 2 3 1 2 3 3 1	1

## Задача Е. Стёпа и Маша

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	64 мегабайта

Стёпе нравится Маша. А Маше нравится Стёпа. Именно поэтому тёплым весенним деньком Маша и Стёпа решили отправиться в парк на романтическую прогулку. Они беззаботно шли по тропинке, держась за руки, как вдруг подошли к странному спортивному снаряду. Он состоял из  $n$  подряд идущих столбиков, расположенных близко друг к другу.

Стёпа быстро оценил высоту каждого из них, и предположил, что высота столбика с номером  $i$  равняется  $a_i$  метров.

Чтобы произвести впечатление на Машу, Стёпа решил выполнить следующее упражнение: он прыгает на столбик с номером 1 и затем  $k - 1$  раз повторяет следующую процедуру: пусть он стоит на столбике с номером  $i$ . Тогда прыгает на столбик  $j$  с минимальным номером, таким что  $j > i$  и  $a_j > a_i$ . Проще говоря, он прыгает на ближайший столбик с большим номером и большей высотой. Если же такого столбика нет, Стёпа теряет надежду заполучить сердце Маши, плачет и уходит домой заниматься дифференциальной геометрией.

Стёпа уже выбрал число  $k$  и подошел к снаряду, как понял, что катастрофически ошибся. Слабое зрение Стёпы подвело его, и он неправильно оценил высоту некоторых столбиков.

— Ничего страшного - подумал Стёпа — и не такое случалось. Если высота этого столбика 16394 метра, то...

И вдруг Стёпа испугался. Он понял, на какую высоту ему придется залезть и понял, что это слишком опасно.

— Я еще так молод — бормотал под нос Стёпа — я впервые влюбился, я только полюбил эту жизнь... И терять её из-за этого снаряда я не готов!

Поэтому Стёпа решил немного уменьшить  $k$ , чтобы так не рисковать.

Но неудачи, казалось, преследовали Стёпу: он то обнаруживал, что высота столбика неверна, то число  $k$  ему казалось неподходящим. Действительно: если он залезет слишком низко, Маша не оценит его способности, а если слишком высоко, есть шанс упасть.

И каждое такое изменение заставляло Стёпу пересчитывать высоту, на которой он в итоге окажется. Казалось, что он будет вечно решать, что же делать, как вдруг Маша крикнула:

— Давай быстрее, милый! Я жду!

Больше откладывать выполнение упражнения было нельзя. Напишите программу, которая будет считать, на какой высоте окажется Стёпа после упражнения.

### Формат входных данных

В первой строке входного файла находятся 2 целых числа ( $1 \leq n, q \leq 500\,000$ ) — количество столбиков и запросов Стёпы, соответственно.

Во второй строке находится  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) разделенные пробелами — начальные оценки высот столбиков Стёпой.

Следующие  $q$  строк содержат запросы Стёпы. Первое число в строке  $t_i$  ( $1 \leq t_i \leq 2$ ) означает его тип.

Если  $t_i = 1$ , то далее следуют два числа  $p_i$   $x_i$  ( $1 \leq p_i \leq n, 1 \leq x_i \leq 10^9$ ) - теперь Степа считаю высоту  $p_i$  столба равной  $x_i$ .

Если  $t_i = 2$ , то далее следуют одно число  $k_i$  ( $1 \leq k_i \leq n$ ) — количество прыжков Стёпы.

### Формат выходных данных

На каждый запрос второго типа выведите высоту на которую заберется Стёпа или '-1' без кавычек, если он не сможет выполнить заданное количество прыжков.

## Пример

стандартный ввод	стандартный вывод
5 13	1
1 2 1 3 5	2
2 1	5
2 2	8
2 4	9
1 1 8	-1
1 3 6	28
1 2 9	-1
2 1	
2 2	
2 3	
1 5 28	
2 3	
1 1 333	
2 2	

## Задача F. Connect

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 6 секунд  
Ограничение по памяти: 256 мегабайт

Дан список из  $m$  ребер, пронумерованных от 1 до  $m$ . Посчитайте количество подотрезков  $[l, r]$  ( $1 \leq l \leq r \leq m$ ) таких, что граф на  $n$  вершинах, содержащий только ребра из этого подотрезка, является связным.

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $m$  ( $2 \leq n \leq 50000$ ,  $1 \leq m \leq 200000$ ) — число вершин и ребер. Следующие  $m$  строк содержат описания ребер. На  $i + 1$ -й строке даны два целых числа  $u_i, v_i$  — концы  $i$ -го ребра ( $1 \leq u_i, v_i \leq n$ ).

### Формат выходных данных

Выведите ответ на задачу.

### Пример

стандартный ввод	стандартный вывод
4 4 1 2 2 4 1 3 1 4	3



## Задача G. Кто прочитал тот сдохнет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Однажды вам приснился кошмар. Вы проснулись и записали условие задачи из сна: Вам дано подвешенное дерево на  $n$  вершинах, у каждой вершины  $v$  есть свой целый вес  $w_v$ . Далее вам поступают  $m$  запросов одного из следующих типов:

- $0\ x\ y$  – присвоить всем вершинам в поддереве вершины  $x$  вес  $y$ .
- $1\ x$  – поменять корень на вершину  $x$ .
- $2\ x\ y\ z$  – присвоить всем вершинам на пути от вершины  $x$  до вершины  $y$  вес  $z$ .
- $3\ x$  – вывести минимум в поддереве вершины  $x$ .
- $4\ x$  – вывести максимум в поддереве вершины  $x$ .
- $5\ x\ y$  – увеличить вес всех вершин в поддереве вершины  $x$  на  $y$ .
- $6\ x\ y\ z$  – увеличить вес всех вершин на пути от  $x$  до  $y$  на  $z$ .
- $7\ x\ y$  – вывести минимум на пути от вершины  $x$  до вершины  $y$ .
- $8\ x\ y$  – вывести максимум на пути от вершины  $x$  до вершины  $y$ .
- $9\ x\ y$  – изменить предка вершины  $x$  на вершину  $y$ , если  $y$  находится в поддереве вершины  $x$  ничего не делать не надо.
- $10\ x\ y$  – вывести сумму на пути от  $x$  до  $y$ .
- $11\ x$  – вывести сумму в поддереве вершины  $x$ .

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ) — число вершин и число запросов соответственно.

В следующих  $n - 1$  строках содержатся по два целых числа, описывающих ребра.

В следующих  $n$  строках содержатся изначальные веса вершин.

Следующая строка содержит корень дерева.

Следующие  $m$  строк содержат запросы в описанном выше формате. Все вершины даны в 1-нумерации. Гарантируется, что все промежуточные вычисления помещаются в `int32_t`.

### Формат выходных данных

Выведите ответы на все запросы, по одному числу в строчке.

## Примеры

стандартный ввод	стандартный вывод
5 5 2 1 3 1 4 1 5 2 4 1 4 1 2 1 10 2 3 3 1 7 3 4 6 3 3 2 9 5 1	9 1 1
10 12 2 1 3 2 4 2 5 3 6 4 7 5 8 2 9 4 10 9 791 868 505 658 860 623 393 717 410 173 4 0 8 800 1 4 2 8 2 103 3 9 4 4 5 7 304 6 8 8 410 7 10 8 8 1 8 9 6 9 10 2 3 11 5	173 860 103 791 608 1557

## Задача Н. Задача для разминки ног

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	512 мегабайт

Однажды мальчик Кирилл долго и упорно пытался решить задачу с говорящим названием «Задача для разминки рук». Получилось ли у него решить задачу мы не знаем, однако, в процессе размышлений Кирилл придумал новую задачу, которую вам и предстоит решить. У Кирилла получилось следующее условие задачи:

В стране  $X$  есть  $n$  городов, соединённых  $m$  дорогами. Множество дорог обладает двумя интересными свойствами. Во-первых, все дороги односторонние. Во-вторых, даже если бы дороги были двусторонними, для любой пары городов существовало бы не более одного простого пути между ними. Нетрудно заметить, что без ориентации граф на данных  $n$  городах в качестве вершин и  $m$  дорогах в качестве рёбер является лесом (то есть в каждой компоненте связности число рёбер на единицу меньше числа вершин).

Функцию  $cost(u, v)$  определим следующим образом: если путь из  $u$  в  $v$  существует, то она равна количеству дорог на этом пути, а если пути нет, то она равна 0.

*Красотой* карты страны назовём величину  $\sum_{u=1}^n \sum_{v=1}^n cost(u, v)$ , то есть сумму величин  $cost$  для всех пар городов.

Правительство страны  $X$  очень любит считать различные статистики. В этот раз было решено посчитать *красоту* текущей карты страны, а так же обработать  $q$  изменений карты страны, после каждого изменения подсчитывая её *красоту*.

Изменения бывают 4 типов:

1. Добавить дорогу из города  $u$  в город  $v$ .
2. Удалить дорогу между городами  $u$  и  $v$ .
3. Изменить ориентацию дороги между городами  $u$  и  $v$  на противоположную.
4. Ориентировать все дороги на пути из  $u$  в  $v$  по направлению из  $u$  в  $v$ . Гарантируется, что это возможно, то есть существует путь из города  $u$  в город  $v$  по дорогам, если игнорировать их ориентацию.

Гарантируется, что после каждого изменения множество городов и дорог продолжает образовывать лес.

Обратите внимание на то, что в данной задаче запросы меняют карту страны и зависят от предыдущих.

### Формат входных данных

Первая строка содержит одно целое число  $g$  ( $0 \leq g \leq 10$ ) — номер группы тестов.

Вторая строка содержит три целых числа  $n$ ,  $m$  и  $q$  ( $2 \leq n \leq 400\,000$ ,  $0 \leq m < n$ ,  $1 \leq q \leq 400\,000$ ) — количество городов, дорог и изменений соответственно.

Каждая из следующих  $m$  строк содержит по два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ) — дороги на изначальной карте ориентированные из  $u$  в  $v$ .

Каждая из следующих  $q$  строк содержит по три целых числа  $t$ ,  $u$  и  $v$  ( $1 \leq t \leq 4$ ,  $1 \leq u, v \leq n$ ) — тип запроса и пару вершин, задающую его.

### Формат выходных данных

В первой строке выведите изначальную *красоту* карты страны.

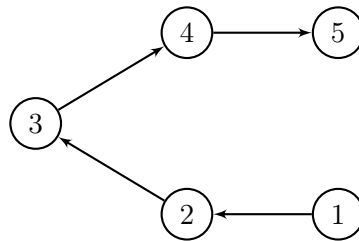
В следующих  $q$  строках выведите по одному числу — значения *красоты* карты страны после каждого изменения.

## Пример

стандартный ввод	стандартный вывод
0	20
5 4 4	6
1 2	3
2 3	4
3 4	16
4 5	
3 4 3	
2 3 2	
1 4 2	
4 1 4	

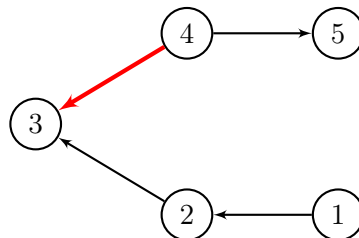
## Замечание

В тестовом примере до всех изменений граф выглядит так:

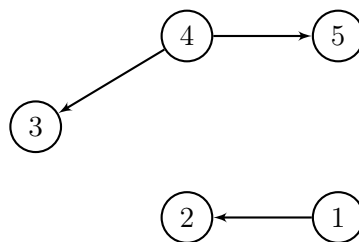


В этом случае, например,  $cost(1,2) = cost(2,3) = cost(3,4) = cost(4,5) = 1$ , так как между этими парами городов есть прямая дорога. Аналогично  $cost(1,3) = cost(2,4) = cost(3,5) = 2$ , так как второй город в каждой паре достижим только за две дороги от первого. Так же, например,  $cost(2,1) = cost(4,1) = cost(5,3) = 0$ , так как второй город в паре не достижим от первого. Суммируя эти величины по всем парам городов, получим 20.

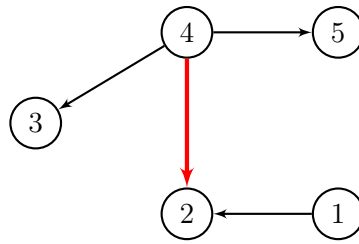
После этого надо изменить направление дороги между городами 3 и 4. Так как до этого она шла от города 3 в город 4, теперь она должна идти от города 4 в город 3. После этого ответ на задачу равен 6, а граф имеет следующий вид:



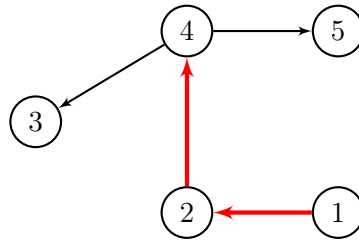
Далее дорога между городами 2 и 3 удаляется. После этого ответ на задачу равен 3, а граф имеет следующий вид:



После этого добавляется дорога из города 4 в город 2. После этого ответ на задачу равен 4, а граф имеет следующий вид:



После этого, все дороги между городами 1 и 4 ориентируются по направлению из города 1 в город 4. После этого ответ на задачу равен 16, а граф имеет следующий вид:



## Система оценки

Тесты к этой задаче состоят из десяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n, m, q$	Типы изменений		
0	0	–	–	–	Тесты из условия.
1	10	$n, m, q \leq 100$	–	0	
2	8	$n, m, q \leq 5000$	–	0, 1	
3	11	$n, m, q \leq 100\,000$	1	–	
4	7	$n, m, q \leq 100\,000$	1, 2	3	
5	13	$n, m, q \leq 100\,000$	3	–	
6	9	$n, m, q \leq 100\,000$	3, 4	5	
7	12	$n, m, q \leq 100\,000$	–	–	$ u-v  = 1$ всегда, кроме запросов 4 типа.
8	18	$n, m, q \leq 100\,000$	–	0 – 7	
9	6	$n, m, q \leq 200\,000$	–	0 – 8	<b>Offline-проверка.</b>
10	6	–	–	0 – 9	<b>Offline-проверка.</b>